

WEB EDİTÖRÜ

8 - JavaScript DOM Erişim Yöntemleri

Öğr.Gör. Serkan KORKMAZ

Nisan - 2023



JavaScript DOM ERİŞİM YÖNTEMLERİ



Belge nesnesi, bir web belgesini veya sayfayı temsil eder.



Bir belgeyi hiyerarşik bir ağaç yapısı olarak temsil eder ve belge ağacındaki her öğeye DOM'da (Belge Nesnesi Modeli) bir Düğüm denir.



Yapısal olarak, Belge Nesnesi Modeli düğümlerden oluşur ve her düğüm web belgesindeki içeriği temsil eder.



Geliştiricilere her şeyi bir web sayfasında göstermek için bir yol sunar böylece web sayfasının içeriğine erişim yöntemler seti aracılığıyla erişilebilir olur.

JavaScript

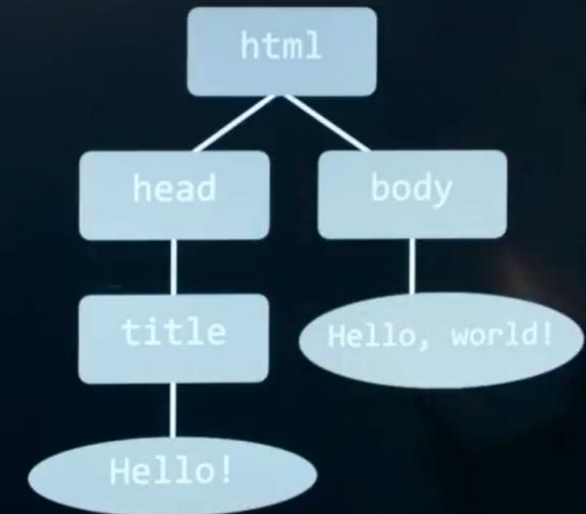
DOM

(Document Object Model)

Belge Nesnesi Modeli

Document Object Model

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello!</title>
  </head>
  <body>
    Hello, world!
  </body>
</html>
```



JavaScript

DOM

(Document Object Model)

Belge Nesnesi Modeli

- Document Object Model, "DOM", "Belge Nesnesi Modeli" anlamlarına da gelmektedir.
- İnternet tarayıcıları girilen internet sitesini bir belge, bu belge (sayfa) içerisinde bulunan tüm elemanları da nesne olarak kabul eder.
- Buna göre resim, yazı, form gibi tüm elemanlar nesnedir.
- İşte DOM sayfa içindeki herhangi bir nesnenin özelliğine müdahale edebilmemize, nesne özelliklerini değiştirebilmemize olanak sağlar.
- Bunu yapabilmek için de JavaScript gibi bazı script dilleri kullanmamız gerekir.

HTML ELEMANLARINI BULMA

- Genellikle JavaScript ile HTML öğelerini değiştirmek isteyebilirsiniz.
- Bunu yapmak için önce HTML öğeleri bulmanız gerekir.
- HTML öğelerini bulmak için JavaScript'de çeşitli erişim yöntemleri sunar.

HTML ELEMANLARINI BULMA

- HTML öğelerini kimlik numarasıyla bulma
- HTML öğelerini etiket adına göre bulma
- HTML öğelerini sınıf adına göre bulma
- CSS seçicileriyle HTML öğelerini bulma
- HTML öğelerini HTML nesne koleksiyonlarıyla bulma



Id ile HTML Elemanı Bulma

Genellikle **DOM** içindeki bir öğeye erişmek ve niteliklerini değiştirmek istiyorsanız. **Javascript**, bir öğeye **DOM** ağaç yapısından erişmenin en kolay yolu olan `document.getElementById()` yöntemini sağlar. Belirtilen değer ile **ID** özneliğine sahip olan öğeyi döndürür.

Aşağıda ki örnek, id = "test" olan öğeyi bulur:

ÖRNEK:

```
var x = document.getElementById("test");
```

Eleman bulunursa, yöntem öğeyi bir nesne olarak (x öğesinde) döndürür. Eleman bulunamazsa, x'de boş değer bulunacaktır.

Etiket Adına Göre HTML Elements Bulma

`getElementsByTagName()` yöntemi, dokümandaki tüm öğelerin belirtilen etiket adına sahip bir koleksiyonunu bir **NodeList** nesnesi olarak döndürür.

NodeList nesnesi düğümlerin bir koleksiyonunu temsil eder. Düğümlere dizin numaraları erişilebilir. Endeks 0'dan başlar.

İpucu: **NodeList** nesnesinin `length` özelliğini, belirtilen etiket adına sahip öğe sayısını belirlemek için kullanabilirsiniz; ardından, tüm öğeler arasında döngü oluşturabilir ve istediğiniz bilgiyi ayıklayabilirsiniz.

İpucu: Parametre "", belgedeki tüm öğeleri döndürür.

ÖRNEK:

```
var x = document.getElementsByTagName("p");
```

Yukarıda ki örnek tüm <p> öğelerini bulur:

ÖRNEK:

```
var x = document.getElementsByTagName("p");
```

Yukarıda ki örnek tüm <p> öğelerini bulur:

ÖRNEK:

Dokümandaki tüm öğeleri alın:

```
var x = document.getElementsByTagName("*");
```

ÖRNEK:

```
var x = document.getElementById("deneme");  
var y = x.getElementsByTagName("p");
```

Yukarıda ki örnek, id = "deneme" öğesini bulur ve sonra "deneme" içindeki tüm <p> öğelerini bulur:

ÖRNEK:

Belgedeki tüm <p> öğelerinin arka plan rengini değiştirin:

```
var x = document.getElementsByTagName("P");  
var i;  
for (i = 0; i < x.length; i++) {  
    x[i].style.backgroundColor = "blue";  
}
```

HTML Öğelerini Sınıf adına göre bulma

`getElementsByClassName()` yöntemi, belgede belirtilen sınıf adına sahip tüm öğelerin bir koleksiyonunu bir **NodeList** nesnesi olarak döndürür.

NodeList nesnesi düğümlerin bir koleksiyonunu temsil eder. Düğümlere dizin numaraları erişilebilir. Endeks 0'dan başlar.

İpucu: Belirtilen sınıf adına sahip öğe sayısını belirlemek için **NodeList** nesnesinin `length` özelliğini kullanabilirsiniz, ardından tüm öğeler arasında döngü oluşturabilir ve istediğiniz bilgiyi ayıklayabilirsiniz.

ÖRNEK:

class="deneme" olan elementi seçin.

```
var x = document.getElementsByClassName("deneme");
```

ÖRNEK:

Hem "test" ve "renk" sınıflarına sahip tüm öğeleri alın:

```
var x = document.getElementsByClassName("test renk");
```

ÖRNEK:

Class = "deneme" olan tüm öğelerin arka plan rengini değiştirin:

```
var x = document.getElementsByClassName("test");  
var i;  
for (i = 0; i < x.length; i++) {  
    x[i].style.backgroundColor = "blue";  
}
```


HTML DOM `querySelector()` Yöntemi

`querySelector()` yöntemi, belgede belirtilen bir CSS seçiciyle eşleşen ilk öğeyi döndürür.

Not: `querySelector()` yöntemi yalnızca belirtilen seçicilerle eşleşen ilk öğeyi döndürür. Tüm eşleşenleri geri getirmek için bunun yerine `querySelectorAll()` yöntemini kullanın.

Seçici, birkaç kez kullanılan belgede bir kimlikle eşleşirse (bir "kimlik", bir sayfada benzersiz olması gerektiğini ve birden fazla kullanılmaması gerektiğini unutmayın), ilk eşleşen öğeyi döndürür.

Sözdizimi :

```
document.querySelector(CSS selectors)
```

Dokümandaki ilk <p> ögesini seçin.

ÖRNEK:

```
document.querySelector("p");
```

<p> ögesinin ilk Class = "test" olan elemanı seçin.

ÖRNEK:

```
document.querySelector("p.test");
```

id="test" olan elemanı seçin.

ÖRNEK:

```
document.querySelector("#test");
```

HTML DOM `querySelectorAll()` Yöntemi

`querySelectorAll()` yöntemi, belgede belirtilen bir **CSS** seçiciyle eşleşen tüm öğeleri statik bir **NodeList** nesnesi olarak döndürür.

NodeList nesnesi düğümlerin bir koleksiyonunu temsil eder. Düğümlere dizin numaraları erişilebilir. Endeks 0'dan başlar.

İpucu: Belirtilen seçiciyle eşleşen öğelerin sayısını belirlemek için **NodeList** nesnesinin `length` özelliğini kullanabilirsiniz, ardından tüm öğeler arasında dolaşıp istediğiniz bilgiyi alabilirsiniz.

NOT : `querySelector` , `jquery` seçicilere göre hızlıdır.

div etiketin alt elemanı p'yi seçin.

ÖRNEK:

```
var x = document.querySelectorAll("div > p");
```

Belgede tüm <h2>, <div> ve öğelerinin arka plan rengini ayarlayın:

ÖRNEK:

```
var x = document.querySelectorAll("h2, div, span");  
var i;  
for (i = 0; i < x.length; i++) {  
    x[i].style.backgroundColor = "green";  
}
```

HTML Elemanlarını HTML Nesne Koleksiyonlarıyla Bulma

Bu örnek, form koleksiyonunda id = "frm1" olan form ögesini bulur ve tüm öge değerlerini görüntüler:

```
var x = document.forms["frm1"];  
var text = "";  
var i;  
for (i = 0; i < x.length; i++) {  
    text += x.elements[i].value + "<br>";  
}  
document.getElementById("demo").innerHTML =
```

Aşağıdaki HTML nesnelere (ve nesne koleksiyonlarına) de erişilebilir:

- `document.anchors`
- `document.body`
- `document.documentElement`
- `document.embeds`
- `document.forms`
- `document.head`
- `document.images`
- `document.links`
- `document.scripts`
- `document.title`
- `document.baseURI`
- `document.cookie`
- `document.domain`
- `document.URL`


```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5 <body>
6
7     Yazı : <br>
8     <textarea rows="10" id="yazi"></textarea>
9     <div id="sonuc"></div>
10
11 <script>
12     var yazi=document.querySelector('#yazi');
13     var sonuc=document.querySelector('#sonuc');
14
15     yazi.oninput=function()
16     {
17         var maksimum=30;
18         this.value=this.value.substr(0,maksimum);
19         var kalan= maksimum-this.value.length;
20         sonuc.innerHTML='Kalan Karakter : '+kalan;
21     }
22 </script>
23
24 </body>
25 </html>
```

Yazı :

Yazı :

Merhaba Dünya

Kalan Karakter : 17

```

1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Hipotenüs Hesaplama</title>
6  </head>
7
8  <body bgcolor="yellow">
9  <h2>JavaScript Hipotenüs Hesaplama</h2>
10 <input placeholder="1. Kenarı girin" type="text" id="txtKenar1">
11 <input placeholder="2. Kenarı girin" type="text" id="txtKenar2">
12 <button id="btn">HESAPLA</button>
13 <div id="sonuc"></div>
14
15 <script>
16 function Hesapla ()
17 {
18     var k1=Number(document.getElementById("txtKenar1").value);
19     var k2=Number(document.getElementById("txtKenar2").value);
20     var hipotenus=Math.sqrt((k1*k1)+(k2*k2));
21     document.getElementById("sonuc").innerHTML="Hipotenüs : "+hipotenus;
22 }
23
24     var hesapBtn=document.getElementById("btn");
25     hesapBtn.onclick=Hesapla;
26
27 </script>
28
29 </body>
30 </html>

```

Hipotenüs Hesaplama

Dosya | C:/Users... | phpMyAdmin | Yazar | Okul | Gelir Gider | G. Akademik

JavaScript Hipotenüs Hesaplama

1. Kenarı girin 2. Kenarı girin HESAPLA

Hipotenüs Hesaplama

Dosya | C:/Users... | phpMyAdmin | Yazar | Okul | Gelir Gider | G. Akademik

JavaScript Hipotenüs Hesaplama

3 4 HESAPLA

Hipotenüs : 5

```
1 <html>
2 <head>
3   <meta charset="utf-8">
4   <title>Üçgen Alanı Hesaplama</title>
5 </head>
6
7 <body bgcolor="yellow">
8   <h2>JavaScript Üçgen Alanı Hesaplama</h2>
9   <input type="text" id="kenar1" placeholder="Taban Uzunluğunu Girin">
10  <input type="text" id="kenar2" placeholder="Yüksekliği Girin">
11  <input type="button" value="Hesapla" id="hesapla">
12 <script language="JavaScript">
13
14
15 function hesapla() {
16   //text kutusu içindeki değerleri okuyup sayi1,sayi2 değişkenine aktarıyoruz.
17   var sayi1=document.getElementById("kenar1").value;
18   var sayi2=document.getElementById("kenar2").value;
19
20   //text kutusuna girilen değerler metinsel türdedir. Hesaplama yapabilmek için sayıya çeviriyoruz.
21   sayi1=Number(sayi1);
22   sayi2=Number(sayi2);
23
24   var alan=sayi1*sayi2/2;
25   alert("Üçgenin alanı:"+alan);
26 -}
27
28 //hesaplama yapması için hesap butonunu seçiyoruz.
29 var hesapBtn=document.getElementById("hesapla");
30
31 //fonksiyonu olaya bağlıyoruz.
32 hesapBtn.onclick=hesapla;
33
34 </script>
35
36 </body>
37 </html>
```

Üçgen Alanı Hesaplama

Dosya | C:/Users... | phpMyAdmin | Yazar | Okul | Gelir Gider | G. Akademik

JavaScript Üçgen Alanı Hesaplama

Taban Uzunluğunu Girin | Yüksekliği Girin | Hesapla

Üçgen Alanı Hesaplama

Dosya | C:/Users... | phpMyAdmin | Yazar | Okul | Gelir Gider | G. Akademik

JavaScript Üçgen Alanı Hesaplama

5 | 10 | Hesapla

Bu sayfanın mesajı

Üçgenin alanı:25

Tamam


```
1 <html>
2 <head>
3   <meta charset="utf-8">
4   <title>Daire Alanı Hesaplama</title>
5 </head>
6
7 <body bgcolor="yellow">
8   <h2>JavaScript Daire Alanı Hesaplama</h2>
9   <input type="text" id="kenar" placeholder="Yarıçapı girin">
10  <input type="button" value="Hesapla" id="hesapla">
11 </script>
12 //fonksiyonu tanımladık.
13 function hesapla() {
14   //text kutusu içindeki değeri okuyup sayı değişkenine aktarıyoruz.
15   var sayi=document.getElementById("kenar").value;
16
17   //text kutusuna girilen değerler metinsel türdedir. Hesaplama yapabilmek için sayıya çeviriyoruz.
18   yaricap=Number(sayi);
19
20   var alan=3.14*yaricap*yaricap;
21   var cevre=2*3.14*yaricap;
22   alert("Dairenin alanı:"+alan+" \n Dairenin çevresi:"+cevre);
23
24 }
25
26 //hesaplama yapması için hesap butonunu seçiyoruz.
27 var hesapBtn=document.getElementById("hesapla");
28
29 //fonksiyonu olaya bağlıyoruz.
30 hesapBtn.onclick=hesapla;
31
32 </script>
33
34 </body>
35 </html>
```

Daire Alanı Hesaplama

Dosya | C:/Users... | phpMyAdmin | Yazar | Okul | Gelir Gider | G. Akademik

JavaScript Daire Alanı Hesaplama

Yarıçapı girin

Daire Alanı Hesaplama

Dosya | C:/Users... | phpMyAdmin | Yazar | Okul | Gelir Gider | G. Akademik

JavaScript Daire Alanı Hesaplama

1

Bu sayfanın mesajı

Dairenin alanı:3.14

Dairenin çevresi:6.28

Tamam

```
1 <html>
2 <head>
3 <meta charset="utf-8">
4 <title>Daire Alanı Hesaplama</title>
5 </head>
6
7 <body bgcolor="yellow">
8 <h2>JavaScript Daire Alanı Hesaplama</h2>
9 <input type="text" id="kenar" placeholder="Yarıçapı girin">
10 <input type="button" value="Hesapla" id="hesapla">
11 <div id="sonuc1"></div>
12 <div id="sonuc2"></div>
13
14 <script>
15 //fonksiyonu tanımladık.
16 function hesapla() {
17 //text kutusu içindeki değeri okuyup sayı değişkenine aktarıyoruz.
18 var sayi=document.getElementById("kenar").value;
19
20 //text kutusuna girilen değerler metinsel türdedir.
21 //Hesaplama yapabilmek için sayıya çeviriyoruz.
22 yaricap=Number(sayi);
23 var pi=3.14; // pi=Math.PI;
24 var alan=pi*yaricap*yaricap;
25 var cevre=2*pi*yaricap;
26 document.getElementById("sonuc1").innerHTML="Dairenin alanı:"+alan;
27 document.getElementById("sonuc2").innerHTML="Dairenin çevresi:"+cevre;
28 }
29
30 //hesaplama yapması için hesap butonunu seçiyoruz.
31 var hesapBtn=document.getElementById("hesapla");
32
33 //fonksiyonu olaya bağlıyoruz.
34 hesapBtn.onclick=hesapla;
35 </script>
36
37 </body>
38 </html>
```

